

Fuzzy Reactive Piloting for Continuous Driving of Long Range Autonomous Planetary Micro-Rovers

A. Martin-Alvarez ¹, R. Volpe, S. Hayati, R. Petras

Autonomy and Control Section
Jet Propulsion Laboratory / California Institute of Technology,
Mail Stop 198-219
4800 Oak Grove Drive,
Pasadena, CA 91109

e-mail: alex@telerobotics.jpl.nasa.gov
Tel: (818) 354-4725 Fax: (818) 393-5007

ABSTRACT

A complete piloting control subsystem for a highly autonomous long range rover will be defined in order to identify the key control functions needed to achieve continuous driving that maximizes the range and number of interesting scientific sites visited during the limited life time of a planetary rover. To achieve continuous driving, a complete set of techniques have been employed: fuzzy based controller; real time artificial intelligence reasoning; fast and robust rover position estimators based on odometry and gyro; efficient stereo vision elevation maps based on grids; and fast reaction and planning for obstacle detection and obstacle avoidance based on simple IF-THEN expert system with fuzzy reasoning. To quickly design and implement these techniques, graphical programming has been used to build a fully autonomous piloting system using just the techniques of classic control concepts of cyclic data processing and event driven reaction. Experimental results using the JPL rover Rocky 7 will be given in order to validate the mentioned techniques for continuous driving.

1. INTRODUCTION: Statement of the Problem and Solution Approach

To reduce cost and operational complexity, as well as increase science data return, a highly autonomous rover is desired in a planetary exploration mission. The human on the ground only needs appropriately abstracted state and status feedback in the telemetry down link. Thus, time delays and temporary loss of communications to Earth are not important because no real time control loops are closed via the up/down link. This also reduces the power consumption of rover subsystems like telecommunication and makes more resources available for the actual locomotion.

A piloting control subsystem (the lowest layer of a hierarchical control architecture) [1] [2] for a Long Range Autonomous Planetary Rover is the key element to achieve the desired autonomy, considering realistic constraints of planetary exploration applications with limited resources in electrical power, computational performance, and communications bandwidth. This piloting control subsystem also considers the capabilities to recover the robot from non-nominal situations such as large tilt of the vehicle, detected hazards, overheating motors, and passed goal condition.

¹ Visiting scientist at JPL

Continuous driving is very desirable for a Long Range Autonomous Planetary Rover to reach goals in a minimum time and thereby maximize the number of interesting scientific sites visited during the limited life time of a planetary rover. More specifically, continuous driving consists of 3 phases: 1) no stopping during the nominal driving when moving to the goal; 2) on the fly obstacle detection; 3) switching between nominal driving and obstacle avoidance. Continuous driving is achieved by simultaneous execution of controllers to servo the rover to goal, estimate rover position, detect hazards, monitor a set of nominal and non-nominal defining functions, and generate possible recovery strategies. Already some efforts have been done in this direction [10] and [11]. However, very little has been done for the validation of a full continuous driving using a realistic planetary rover prototype.

A successful implementation requires robust and fast control algorithms, fast communication among control functions, robust and fast position estimation, fast access to a low-memory-usage representation of the environment, and an adequate combination of reactive and fast planning/decision making techniques.

Section 2 describes the guidelines for the design of the piloting control layer for a Long Range Scientific Mars Rover following the principles of Mobile Robot Control Architecture (MORCA) [1] consisting mainly in the definition of three parallel functional branches: forward control, nominal feedback, and non-nominal feedback. Graphical programming is presented for the whole design of the piloting system, supporting the classic control concepts of cyclic data processing and event-driven reaction to achieve all the reasoning and behaviors needed. For this purpose, a commercial graphical tool is presented that includes the mentioned control capabilities. Also efficient message queues are used for fast inter-communication among control functions, allowing real time Artificial Intelligence (AI) reasoning techniques [13] based on queue manipulation.

Section 3 describes the piloting nominal feedback, and in more detail an advanced filter for position estimation based on the combination of odometry and angular rate sensor data. Also a discrete environment representation for a stereo vision elevation map [8] is presented as a simple and efficient environment representation.

Section 4 describes the piloting forward control, where a simple and fast fuzzy controller is used to work in real time, handling non-linearities and eliminating the additional complexity of having a trajectory generator.

Section 5 describes the piloting non-nominal feedback which consists of a fast obstacle detector and a fast obstacle avoidance strategy, implemented as simple expert systems with fuzzy reasoning over the discrete environment representation.

Finally, Section 6 gives experimental results of continuous driving using the JPL planetary rover prototype Rocky 7 in the JPL MarsYard test facility.

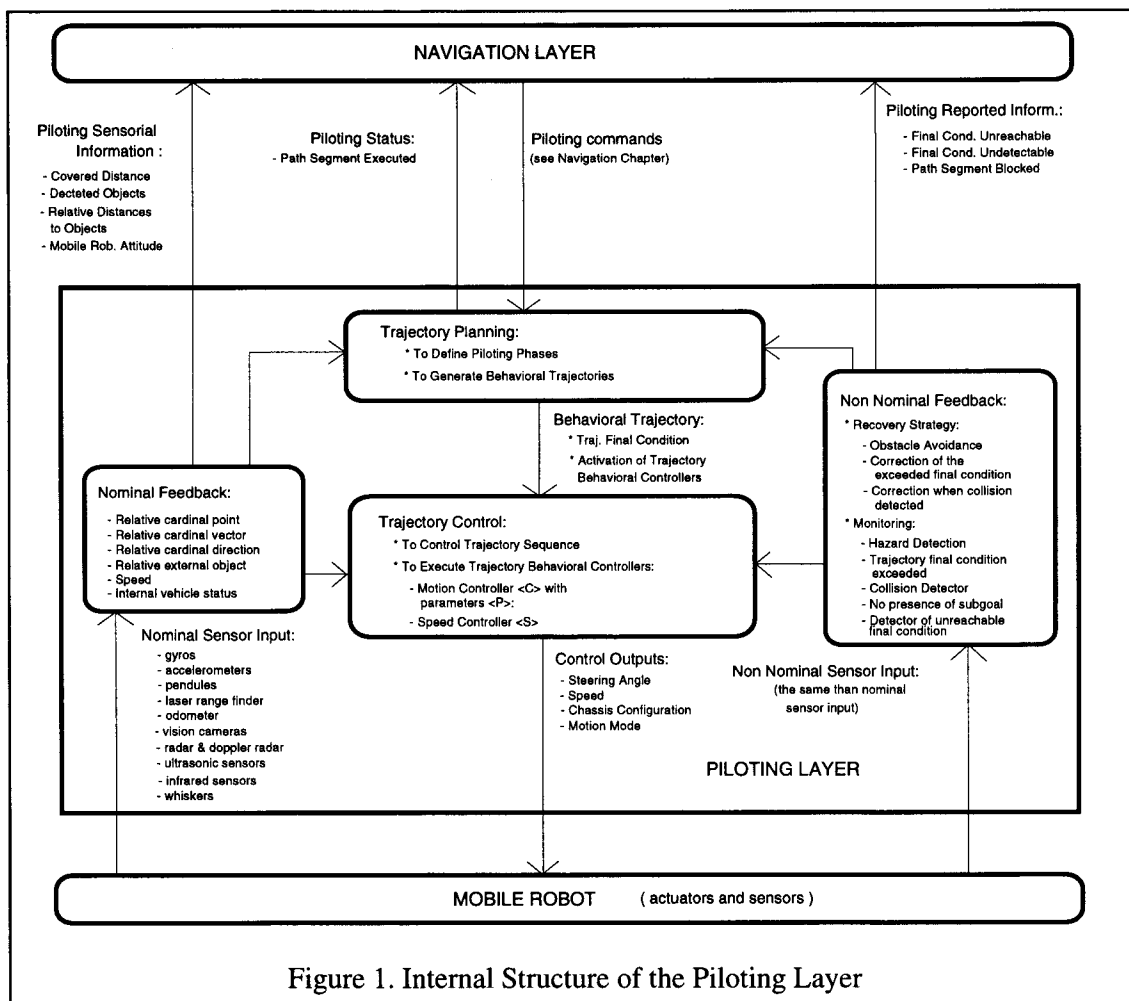


Figure 1. Internal Structure of the Piloting Layer

2. MORCA. PILOTING LAYER

The control functions and structure of an autonomous piloting system for a Long Range Planetary Rover are identified using the principles of Mobile Robot Control Architecture (MORCA) [1][2]. MORCA architecture corresponds to a subsequent refinement of commands to the mobile robot, from highest level mission commands via commands on navigation, piloting, wheel motion coordination, to individual wheel control commands. The decomposition and the control of such commands are performed by a set of functions contained in a control architecture. Because of the mentioned hierarchy in mobile robot commands, MORCA also follows a hierarchical structure, based on different layers where each layer is structured into three parallel functional branches as shown in Fig. 1 [7]. These branches are:

- **Forward Control (FC).** Responsible for task decomposition, execution planning, and control.
- **Nominal Feedback (NF).** Functions for refinement and update of a priori knowledge ("world models") based on the actual, but essentially expected, evolution of the process and consequently formulation of controlled adjustments of the FC.
- **Non-Nominal Feedback (NNF).** Functions for the monitoring of discrepancies between actual and allowable states in both the FC and the NF functions, diagnosis of their origins, and generation of directives and constraints for FC.

The piloting control subsystem is the lowest layer of the hierarchical architecture MORCA receiving commands from a path planner and sending signals to actuators like motor position and motor speed. A complete set of path planner commands were identified in [3]: MOVE_TO waypoint, FOLLOW cardinal direction; BORDER landmark; REACH landmark; and CLIMB slope.

Following the hierarchical engineering approach for an efficient design and real time execution, the pilot layer has been refined into three sub-layers for the control of body motion, wheel co-ordination, and single wheel motion.

In order to achieve real time execution, we follow the control philosophy of combining reactive and planning techniques [12] consisting mainly of body motion planning which activates one or several body motion controllers. These body motion controllers are [1]: "move to cardinal point", "move to cardinal vector", "reach external object", "border external object", "climb slope", "keep direction", "keep speed", and "up/down body".

We have designed message queues and a complete set of queue manipulation procedures that are common for the inter-communication among all the piloting control functions. The two main types of queues that have been implemented are FIFOs (First Input First Output) and LIFO (Last Input First Output). FIFO's queues are the most commonly used mainly for both the execution of sequence of commands and the processing of nominal feedback information in order of occurrence. LIFO's queues are mainly used for non-nominal event communication since the most recent non-nominal event has to be processed immediately in order to evaluate its priority of execution. In order to promote simplicity and efficiency in achieving the high degree of autonomy and intelligence required, it was found convenient to use the same queues and queue manipulation procedures for Artificial Intelligence (AI) reasoning techniques based on queue manipulation [13] like adding, inserting, deleting, and modifying set of commands or recovery strategy directives.

The design of the piloting system consists of the use of classical control concepts of cyclic data processing and event-driven reaction to achieve all the reasoning and behaviors needed. A Finite State Machine technique is used for event-driven reactions, for nominal planning, dispatching, non-nominal diagnosis, generation of non-nominal recovery strategies, and external event handling. On the other hand, synchronous dataflow software is used for the piloting controllers like body servoing and motor control, nominal feedback like position estimation, and non-nominal monitors like obstacle detection.

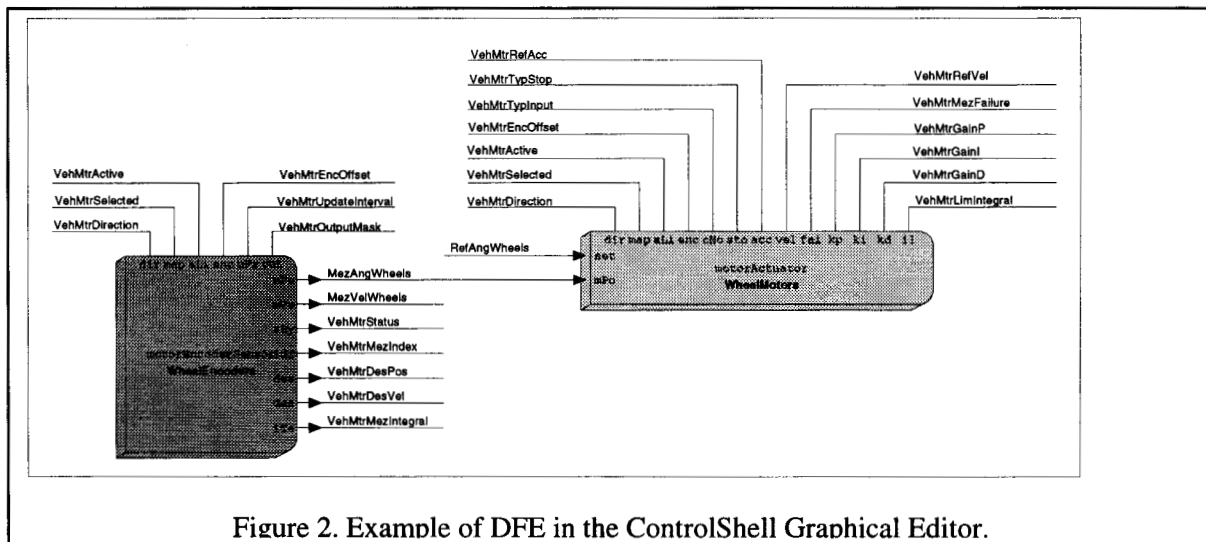


Figure 2. Example of DFE in the ControlShell Graphical Editor.

We use a graphical programming software tool for the full design of our piloting control subsystem. In order to reduce cost, minimize design time, and promote software integration as part of teamwork, the commercial graphical tool ControlShell was chosen [9]. Figure 2 and 3 show how a Data flow diagram and a Finite State Machine are represented in the ControlShell tool. In addition ControlShell provides system configuration control for changing operating modes and real-time

matrix mathematics package called CSMat, where names/labels are placed on the data flow lines and arrows (see Fig. 2), useful for the mentioned real time AI reasoning based on queue manipulation.

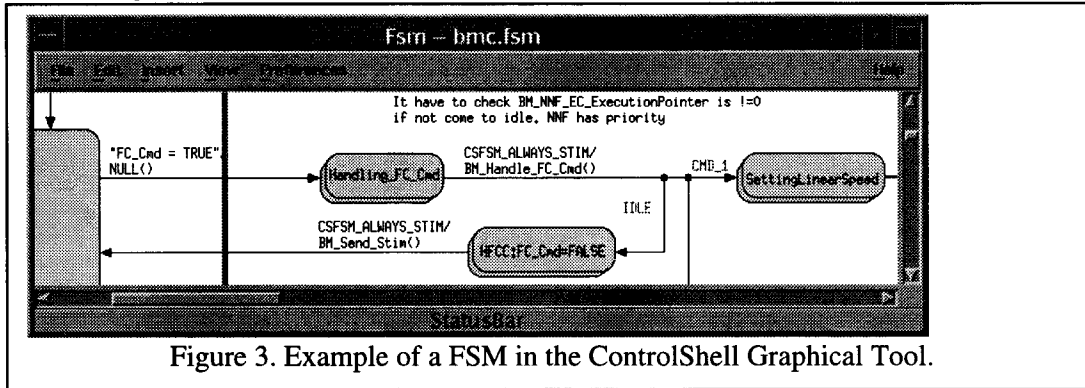


Figure 3. Example of a FSM in the ControlShell Graphical Tool.

3. NOMINAL FEEDBACK. POSITION ESTIMATION

As mentioned before, all the nominal feedback functions are represented as data flow diagrams (see Fig. 2) and are considered as cyclic data processing. A key nominal feedback function in the planetary rover continuous driving is position estimation.

A piloting position estimator suitable for continuous driving has been developed to overcome the challenging requirements of: robustness in the presence of sensor inaccuracies, sensor noise, and undesirable physical rover behaviors like slipping and skidding.

This position estimator first determines the distance traversed using the optical encoders of the wheels and then calculates the heading based on both the optical encoders and gyro. The main feature of our position estimator is to give a good estimation of the rover heading, the main cause of error for any position estimator. For this reason, we designed a simple, robust, and reliable heading estimator merging the best performance regimes of the encoder-based odometer and gyro. First, heading based on odometry is good for smooth and slow turns, whereas gyro measurements are not reliable because the noise and signal cannot be distinguished easily. Second, the heading estimation based on the gyro is excellent for fast turns where the signal can be well differentiated from the sensor noise but odometry fails because of the wheel slippage and wheel skidding. Figure 4 shows on the top the noisy angular speed measured directly from the gyro sensor. On the bottom is the heading estimation based on the integration of the measured angular speed only for fast turns. See Figure 5 as an example of heading estimation at 10 Hz.

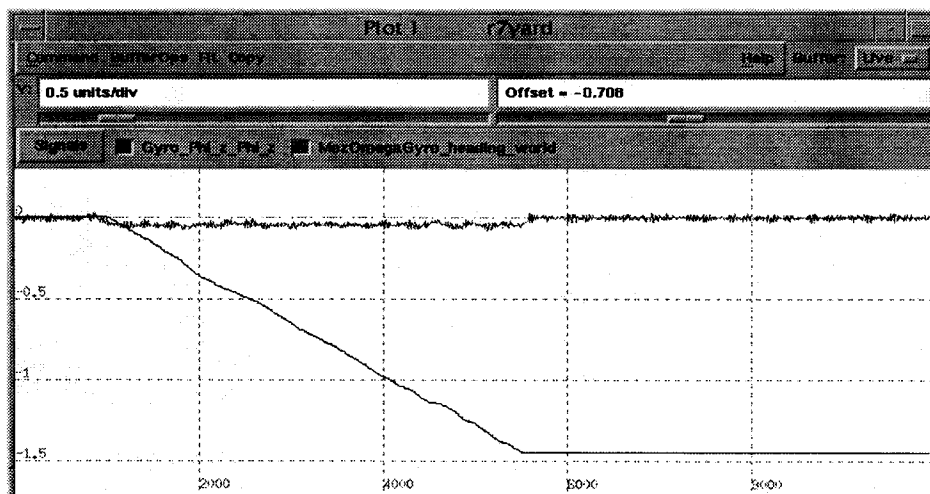
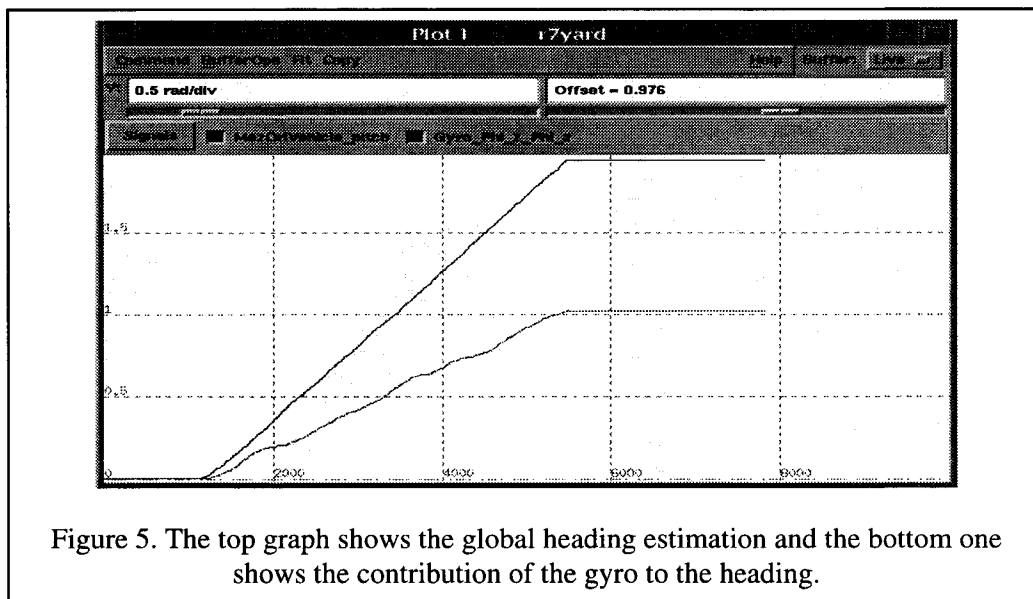


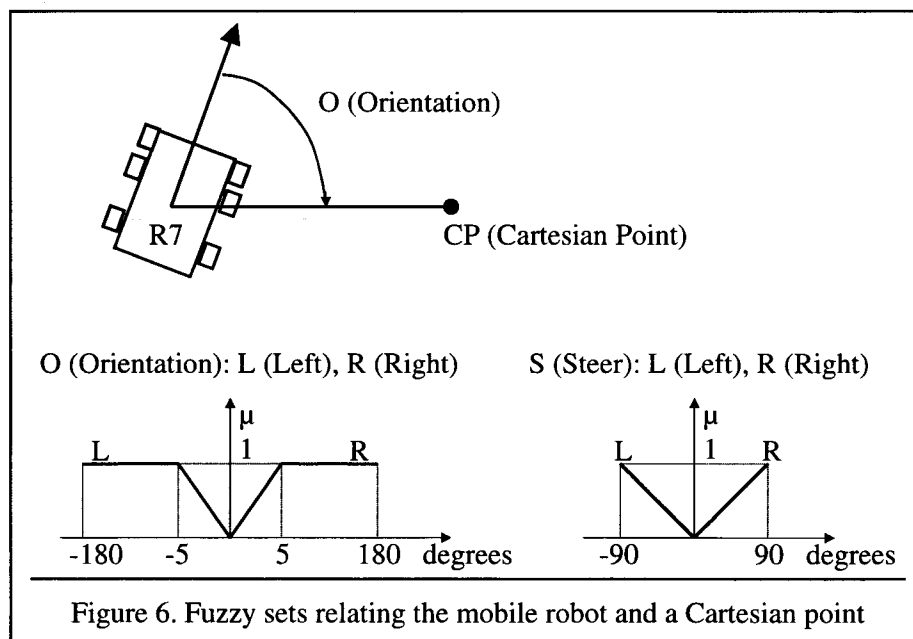
Figure 4. On the top is the noisy angular speed measured directly from the gyro sensor. On the bottom is the heading estimation based on the integration of the measured angular speed including sensor bias compensation.



4. FORWARD CONTROL. FUZZY CONTROLLER

For continuous driving, fuzzy logic based controllers are chosen [17] [18] [19] for their simplicity, ease of understanding, design, and implementation, fast execution, and ability to easily deal with non-linearities and constraints.

Figure 6 shows the fuzzy sets of the fuzzy control variables for the controller "go to a way point". the control variable "Orientation" indicates the direction where the goal, that is, the Cartesian goal point is located with respect to the rover main axis. Two fuzzy sets have been defined for "Orientation": L (Left) and R (Right). Each "Orientation" value from -180 dg to 180 dg has a degree of membership to the fuzzy sets L and R.



The fuzzy controller "go to a way point" consists of two rules of type "IF-THEN" commanding the rover to turn left if the goal is to the left, and turn right if the goal is to the right (see Figure 7). This fuzzy controller can be seen in classical control terminology as a proportional controller for the range of "Orientation" between -5 and 5 degrees with saturation (non-linearity) outside this range.

Controller 1 : Go to a Cartesian Point, <Forward / Backward>**R1: IF Orientation(Left) THEN Steer (Left)****R2: IF Orientation(Right) THEN Steer (Right)**

Figure 7. Fuzzy Controller Rules for the Controller to go to a Cartesian Point

Similar representation are used for other more complex piloting fuzzy controllers [1] like "go to a Cartesian vector", "follow a Cartesian line", and "border a rock". However one of the major advantages of using just a fuzzy controller is that no trajectory planning is needed for continuous driving. The reduced complexity frees the CPU for the execution of other functions including the high time consuming stereo vision algorithms.

5. NON-NOMINAL FEEDBACK.

The fast and robust execution of the non-nominal functions for obstacle detection and of generation of an obstacle avoidance strategy are mandatory for continuous driving. Obstacle detection and obstacle avoidance strategy generation is based on the criteria of cell occupied or free of obstacle.

5a. Obstacle Detection

Another key function in the non-nominal feedback is the environment representation. Again, following the design principle of maximum simplicity, a stereo vision elevation map (see Fig. 8) based on a grid of 1.5 m long x 1 m wide (see Fig. 9) has been used to represent the piloting environment measured in front of the rover. Each cell of the grid is labelled considering its distance to the rover and orientation with respect to the rover main axis.

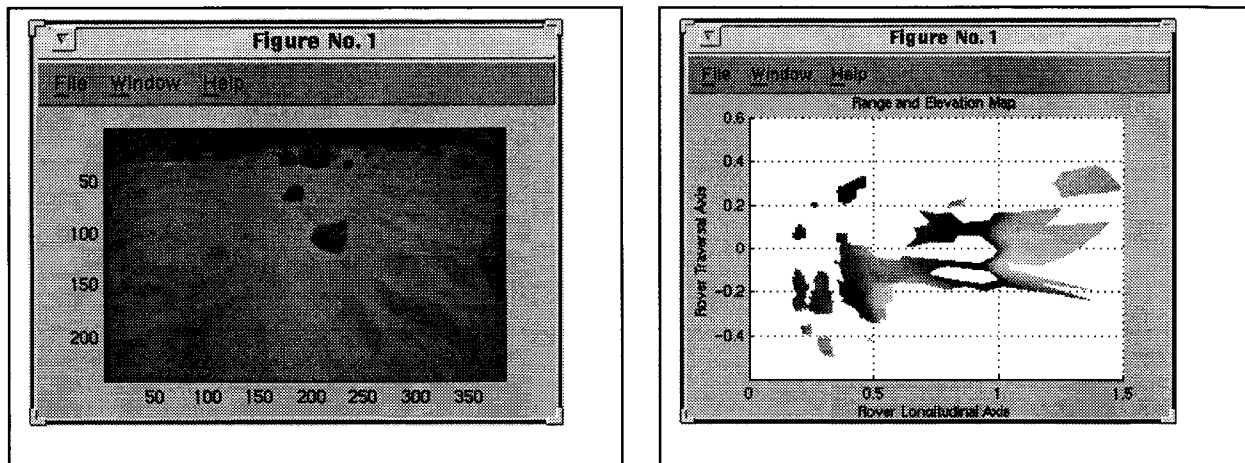
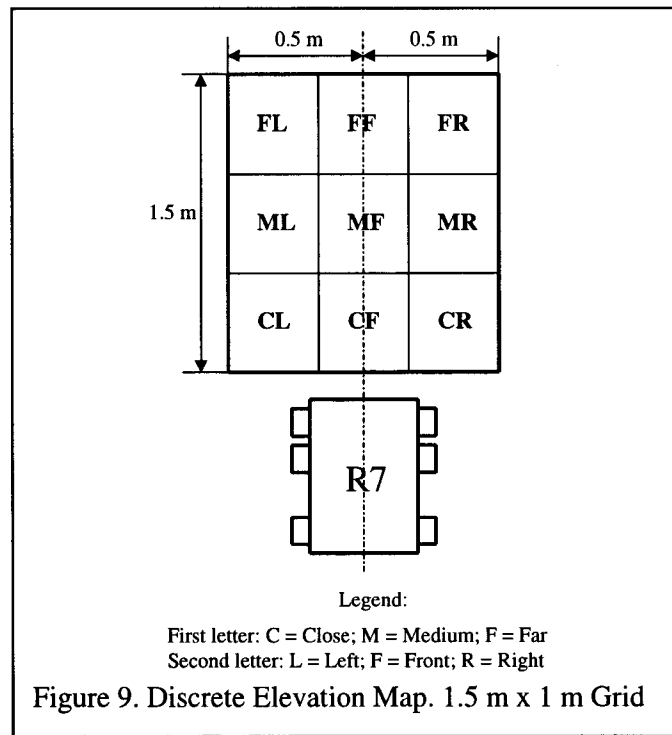


Figure 8. Picture and its respective Elevation Map using the pair of stereo cameras mounted for obstacle detection and obstacle avoidance mounted under the solar panel (see Fig. 10). The darker the areas in the Elevation Map the higher they are. White areas mean no stereo data.

A cell is selected as occupied when the number of high elevations is bigger than a predefined value, used as a filter threshold, that indicates the existence of one or several obstacles.

The obstacle detection is done when one or several occupied cells are close enough to be considered as a hazard, that is, obstacle avoidance manoeuvre cannot be performed without collision.



5b. Obstacle Avoidance

Once an obstacle has been detected, a simple and fast expert system as a set of IF-THEN rules with fuzzy reasoning will make the decision to avoid the obstacle either on the left or on the right.

Basically the obstacle avoidance strategy is generated when the first of the following three ordered rules is true:

1st rule: If the obstacle is very close then backup a bit to avoid collision.

2nd rule: If the obstacle is not very close then no collision is assured and search for the first free cell to go scanning the map from right to left and from bottom to up.

3rd rule: If there are no free cells then select MR (Medium Right) cell by default.

Once a cell is selected, the obstacle avoidance strategy consists of two basic turn manoeuvres to direct the rover outside of the selected cell. For example, if a cell on the right was selected, then first a small turn to right and later to the left is executed to face the rover to a hypothetical cell on the right side of the selected cell. If an obstacle is detected again, the same obstacle avoidance procedure is repeated, otherwise the fuzzy controller will steer the rover to the goal.

6. EXPERIMENTAL RESULTS. CONTINUOUS DRIVING.

The JPL-NASA micro-rover Rocky 7 [4][5][6] was used as test vehicle for the testing of continuous driving of a Long Range Autonomous Space Rover. Rocky 7 (see Fig. 11) is a research micro-rover used to demonstrate new technology concepts for use in a long range (>1 Km) traversal across Mars, scheduled for early in the next decade. Its locomotion is a modified six-wheel rocker/bogey similar to Sojourner (NASA PathFinder mission) [16].

Its main features are: 1) size: 60 x 40 x 33 cm; 2) mass: 15.7 kg; 3) power: rechargeable NiCad batteries and Si solar panel; 4) computer: 3U VME, 68060 CPU, 100 MIPS.



Figure 10. JPL-NASA Micro-Rover Rocky7 with its mast up

Obstacle detection and obstacle avoidance are done using the pair of stereo cameras mounted under the solar panel, effectively ranging from the edge of the rover up to 1.5 meters. This range together with the stereo vision system processing time (8 seconds in total) defined the safest speed of 10 cm/s for our experiments.

Figure 11 shows an example of the trajectory followed by the rover using only the fuzzy controller to go from its initial position (0,0) with 0 heading (on the positive semi-axis x) to the goal position (1,1) meters. It can be seen how the fuzzy controller strongly steers the rover at the beginning in order to point it to the goal. Also it can be observed the robustness of the fuzzy controller steering the rover to the overpassed goal position when the detector of final condition was intentionally deactivated.

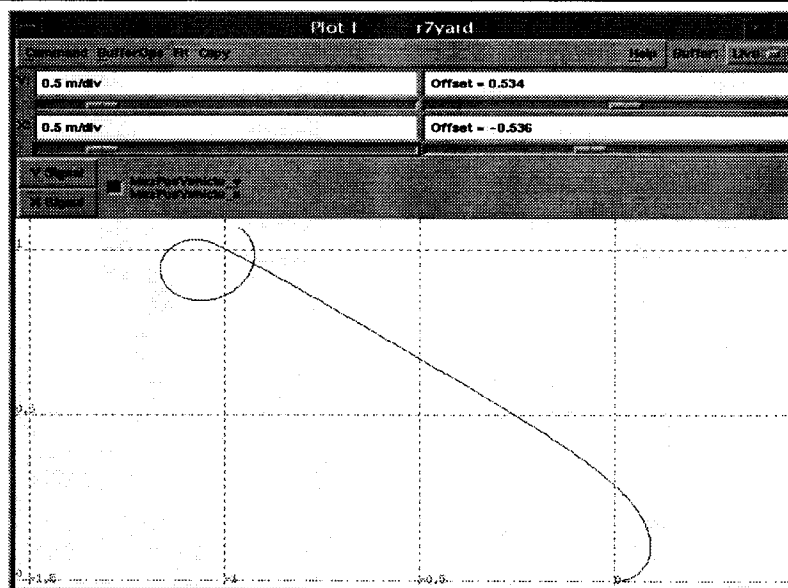


Figure 11. Rover Trajectory from the (0,0) m to (1,1) m with final condition deactivated

Figure 12 shows the testing scenario and more specifically the JPL Mars Yard seen from the Rocky 7's mast cameras. Two major obstacles can be seen clearly in front of the rover. The test scenario was to command the rover to traverse 100 m in a given direction. The objective was to traverse and avoid obstacles without stopping the rover.

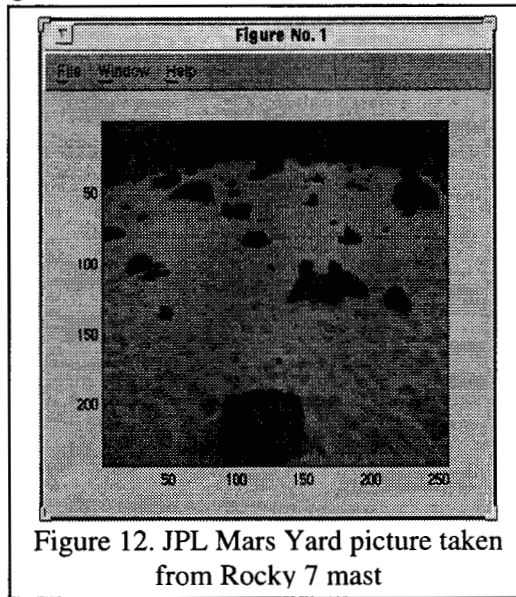


Figure 12. JPL Mars Yard picture taken from Rocky 7 mast

Figure 13 shows the traversed rover trajectory from its initial position at (0,0) to (6,0) meters. First the rover moves forward as the fuzzy controller is steering the rover to the goal (100,0) meters then an obstacle is detected after traversing 1.5 meters, and the obstacle avoidance strategy of two turns is generated to avoid the obstacle on the right side. After this obstacle avoidance manoeuvre there was no obstacle in front of the rover, and the fuzzy controller steers the rover to the goal. It can be seen that the fuzzy controller does not steer the rover to follow the x-axis as might be expected if a simple trajectory generator was used. Instead the fuzzy controller points the rover again to (100,0) and almost a parallel trajectory to the X axis was followed as the goal is still very far away.

A second obstacle is found after traversing 4.2 meters. In this case, the detected obstacle was very close to the rover because it the obstacle was smaller than the previous one. The first obstacle avoidance movement is to backup a bit (small straight-line from (4.2, -0.75) meters to (3.8, -0.75) meters). The obstacle was on the right, hence causing the rover to avoid it by turning to the left. As before, the fuzzy controller points the rover again to the goal.

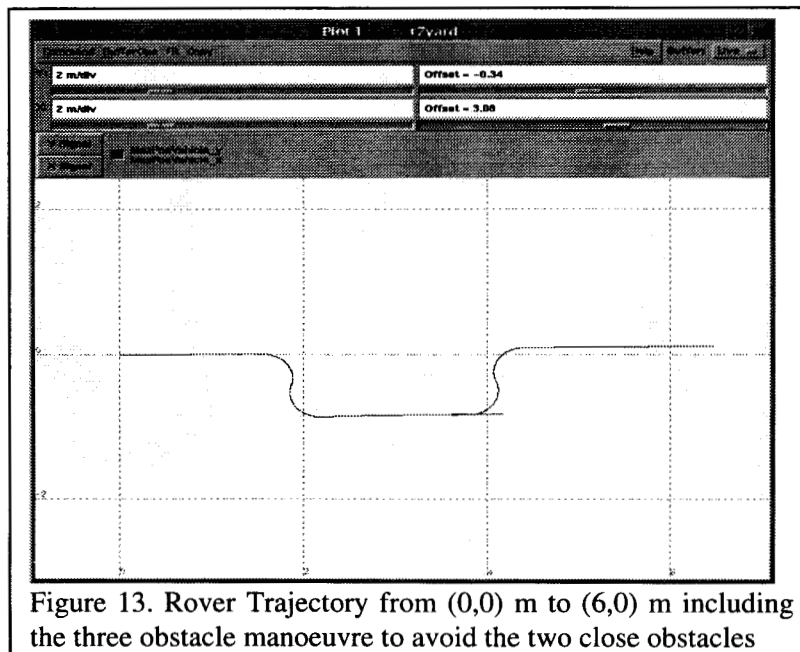


Figure 13. Rover Trajectory from (0,0) m to (6,0) m including the three obstacle manoeuvre to avoid the two close obstacles

The position estimator based on both optical encoders and gyro has been proved to be suitable for continuous motion with a small error percentage for normal piloting traverses of approximately 20 meters without the need to stop for a global position estimation based on the known position of landmarks and celestial bodies.

CONCLUSIONS

Most of the key control functions of an autonomous piloting system, that has to deal with nominal and non-nominal situation, have been identified for a Long Range Planetary Rover using the principles of Mobile Robot Control Architecture (MORCA). To achieve continuous driving, we have defined the following piloting techniques: a robust and fast fuzzy logic controller; fast inter-communication among control functions based on queues that at the same time are used for real time Artificial Intelligence (AI) reasoning techniques; fast and robust rover position estimation based on odometry and gyro measurement; a low memory and fast access environment representation as a grid based stereo vision elevation map suitable for fast nominal and non nominal decision making; and a fast reaction and planning obstacle detection and obstacle avoidance scheme based on simple IF-THEN fuzzy reasoning. Using these techniques experimental results using the JPL Rocky 7 rover have been shown to validate the concept of continuous driving for future planetary rovers.

ACKNOWLEDGMENT

Most of the research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, in the Long-Range Science Rover Project under contract with the National Aeronautics and Space Administration. The first author of this paper wants to thank Dr. Samad Hayati and the Spanish Government, and more particularly El Ministerio de Educacion y Ciencia Subprograma de Perfeccionamiento de Tecnologos y Doctores en el Extranjero, for his research fellowship that allowed him to continue their research in Space Rovers at JPL.

BIBLIOGRAPHY

- [1] A. Martin-Alvarez, "Piloting and Navigation of Mobile Robots for Planetary Explorations". Doctoral Thesis, Universidad Politecnica de Madrid/ Escuela Tecnica Superior de Ingenieros Industriales (in Spanish), March 1995.
- [2] A. Martin-Alvarez, W. De Peuter, P. Putz, " A Unified Control Architecture for Planetary Rovers". Artificial Intelligence, Robotics, and Automation for Space "i-SAIRAS 94". Jet Propulsion Laboratory, Pasadena, California, USA. October 18-20, 1994.
- [3] A. Martin-Alvarez "A Fuzzy Pilot for a Microrover", 2nd International Symposium Mission, Technologies and Design of Planetary Rovers, Moscow 94.
- [4] S. Hayati, R. Volpe, et al., "The Rocky 7 Rover: A Mars Science Spacecraft Prototype". Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque NM, April 20-25 1997.
- [5] Rocky 7 home page: "<http://robotics.jpl.nasa.gov/tasks/scirover/homepage.html>". Maintained by R. Volpe.
- [6] R. Volpe, J. Balaram, T. Ohm, and R. Ivlev. The Rocky 7 Mars Rover Prototype. In IEEE//RSJ International Conference on Robots and Systems (IROS), Osaka, Japan, November 4-8 1996.
- [7] P. Putz, A. Elfving, "A Development Methodology for Space A&R Control Systems". 12th IFAC Symp. on Automatic Control in Aerospace (Aerospace Control 92). Ottobrunn, Sept. 7-11, 1992.
- [8] L. Matthies and P. Granjean "Stochastic Performance Modeling and Evaluation of Obstacle Detectability with Imaging Range Sensors". IEEE Transaction on Robotics and Automation, 10(6):783-791, December 1994.

- [9] S. A. Schneider, "Extending ControlShell for Intelligent Reactive System". Phase II Final Report. SBIR NASA Contract (NAS7-1391).
- [10] E. Gat, M.G. Slack, D.P. Miller, R.J. Firby. "Path planning and execution monitoring for a planetary rover". Proc. of the IEEE Int. Conf. on Robotics and Automation, May 90.
- [11] B. H. Wilcox, "A vision system for a Mars rover". Proceedings of SPIE Mobile Robots II, vol. 852, November 1987.
- [12] D. W. Payton et al., "Plan Guided Reaction". IEEE Transactions on Systems, Man, and Cybernetic. Vol 20, No 6, November/December 1990.
- [13] N. Nilsson, "Principles of Artificial Intelligence", Toga Publishing.
- [14] B. Barshan and H. F. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots". IEEE Transactions on Robots and Automation, Vol. 11, No. 3, June 1995.
- [15] Y. Fuke and E. Krotkov, "Dead Reckoning for a Lunar Rover on Uneven Terrain". IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996.
- [16] H. W. Stone, "Mars Pathfinder Microrover A Low-Cost, Low-Power Spacecraft". In Proceedings of the 1996 AIAA Forum on Advanced Developments in Space Robotics, Madison, WI, August 1996.
- [17] L. A. Zadeh, "Fuzzy Sets, Information and Control", 8, 1965, 338-353.
- [18] M. Sugeno and M. Nishida, "Fuzzy Control of Model Car". Fuzzy Sets and Systems 16: 103-113.
- [19] E. H. Mamdani, "Application of fuzzy algorithms for the control of a simple dynamic plant". Proceedings of IEE 121: 1585-1588, 1974.